Continue to work with your partner on this assignment. Don't forget to switch roles often.

Exercise 2.0: Unblocking

Complete the following private helper `removeBlocks` method in your `Tetrad` class.

```
// precondition:  Blocks are in the grid.
// postcondition: Returns old locations of blocks;
//                blocks have been removed from grid.
private Location[] removeBlocks()
```

Run `TetradTest` to verify that you have completed this exercise before moving on.

Exercise 2.1 Anybody Home?

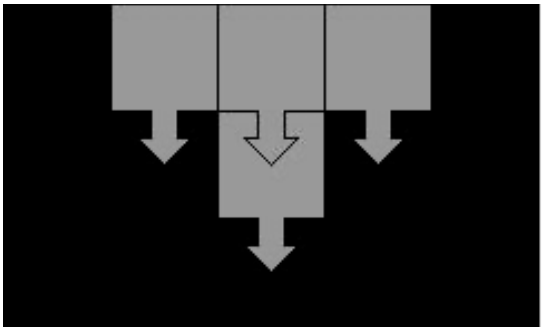
Complete the following private helper `areEmpty` method in your `Tetrad` class.

```
// postcondition: Returns true if each of locations is
//                valid (on the board) AND empty in
//                grid; false otherwise.
private boolean areEmpty(Grid grid, Location[] locations)
```

Run `TetradTest` to verify that you have completed this exercise before moving on.

Exercise 2.2: Lost in Translation

We'll now complete the `Tetrad` `translate` method, which will shift the tetrad over and down by the given amount, first making sure that the tetrad's potential new position is valid and empty. For example, suppose we have a T-shaped tetrad in the top middle of an empty grid, and we wish to move it down by one row, as shown below.



Clearly, we ought to allow this move. But notice that one of the locations we're attempting to move this tetrad into is already occupied by another block in the tetrad itself. So, it's not enough to make sure that the new locations are empty.

To make sure we handle this situation correctly, we'll always translate a tetrad as follows:

1. Ask any block for its grid, and store it in a temporary variable.
2. Remove the blocks in the `Tetrad` (temporarily saving the old locations).
3. Create an array of the new (possible) locations.
4. Check if the new locations are empty (and valid).
5. If the new locations are empty, add the `Tetrad` to the new locations, and return true. Otherwise, add the `Tetrad` back to its original locations and return false.

Go ahead and complete the `Tetrad` `translate` method, making use of the helper methods `addToLocations`, `removeBlocks`, and `areEmpty`.

```
//postcondition: Attempts to move this tetrad deltaRow
//               rows down and deltaCol columns to the
//               right, if those positions are valid
//               and empty; returns true if successful
//               and false otherwise.
public boolean translate(int deltaRow, int deltaCol)
```

Run `TetradTest` to verify that you have completed this exercise. Please submit a screenshot of the tester success message for Part - 2.